

## Desafio – Área de qualidade

Você deve testar um aplicativo desenvolvido para calcular o dígito verificador de uma série de matrículas da empresa XPTO.

Dada uma matrícula de **0000** a **9999** o cálculo do dígito verificador deverá ser feito multiplicando o primeiro número à esquerda por **5**, o segundo por **4**, o terceiro por **3** e o quarto por **2**.

O resultado parcial de cada posição deve ser somado e o resultado final da soma deverá ser dividido por **16**. O resto desta divisão deverá ser convertido para hexadecimal, sendo este então o dígito verificador.

Exemplo:

<b>Matrícula</b>
9876
<b>Cálculo</b>
$9 * 5 = 45$
$8 * 4 = 32$
$7 * 3 = 21$
$6 * 2 = 12$
<b>Total</b> = $45 + 32 + 21 + 12 = 110$

Resto da divisão de **110** por **16** é igual a 14.

O número **14** em base **16** é "**E**". Este é o dígito verificador da matrícula.

Matrícula completa = **9876-E**

O programa deve ler as matrículas que estão em um arquivo texto de entrada e gerar um arquivo texto de saída com o nome de "matriculasComDV.txt" no mesmo diretório do arquivo de entrada. O arquivo de saída deve conter as matrículas completas, conforme regra de formação descrita acima.

Exemplo:

entrada.txt	matriculasComDV.txt
9876	9876-E
9992	9992-0

O que DEVE ser enviado como resultado deste desafio:

1. Documento contendo os casos de teste.
2. Documento contendo os bug reports (tickets) criados a partir dos testes executados na aplicação. A ideia é demonstrar como as questões encontradas (bugs, ajustes, melhorias) poderiam ser reportadas.

Os itens abaixo não são obrigatórios, mas podem ser um diferencial:

3. Implementação de um conjunto de testes automatizados para a aplicação em questão.
4. Documento contendo os passos necessários para a instalação e execução dos testes automatizados implementados.

### Importante:

- Todos os arquivos devem ser enviados **em um único ZIP**.
- Não incluir os arquivos binários (.exe, .jar, .dll) no ZIP pois alguns serviços de e-mail bloqueiam este tipo de arquivo.